

DOCUMENT RESUME

ED 247 289

TM 840 469

AUTHOR Peay, Edmund R.
TITLE Constructing Questionnaires with Randomized Ordering and Sampling of Items: A Computer Implemented Method. CASSR Technical Paper Series, No. 4.
INSTITUTION Flinders Univ. of South Australia, Bedford Park. Centre for Applied Social and Survey Research.
REPORT NO ISBN-O-7258-0201-4
PUB DATE 82
NOTE 41p.
AVAILABLE FROM Centre for Applied Social and Survey Research, School of Social Sciences, The Flinders University of South Australia, Bedford Park, S.A., 5042, Australia (\$3.50 plus postage).
PUB TYPE Reports - Descriptive (141) -- Computer Programs (101)
EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Computer Assisted Testing; Computer Software; *Questionnaires; Sampling; *Test Construction; Test Format; Test Items

ABSTRACT

The method for questionnaire construction described in this paper makes it convenient to generate as many different forms for a questionnaire as there are respondents. The method is based on using the computer to produce the questionnaire forms themselves. In this way the items or subgroups of items of the questionnaire may be randomly ordered or sampled as each form is being generated. This provides a very direct way of dealing with order and combination effects. The technique also provides a number of incidental benefits, such as direct production of forms bypassing some stages of the usual process, and the prevention of collaborative responses in a group administration situation, since each form is "different". The paper includes a description of the method and of a pair of computer programs designed to implement it, a User's Guide for each program, and program listings. (Author/BW)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

ED247289

EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)
☒ This document has been reproduced as
received from the person or organization
originating it.
☐ Minor changes have been made to improve
reproduction quality.
• Points of view or opinions stated in this docu-
ment do not necessarily represent official NIE
position or policy.

CASSR

TECHNICAL PAPER SERIES No. 4

CONSTRUCTING QUESTIONNAIRES WITH RANDOMIZED ORDERING AND SAMPLING OF ITEMS: A COMPUTER IMPLEMENTED METHOD

Edmund R. Peay

Senior Lecturer in Psychology
The Flinders University of South Australia

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

E. R. Peay

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)."

7/11/84 0.469

**The Centre for Applied Social and Survey Research
Technical Papers Series.**

The Centre for Applied Social and Survey Research (CASSR) is a multi-disciplinary research Centre established in the School of Social Sciences at the Flinders University of South Australia in 1977. It undertakes research in the areas of urban and regional studies, community development, human behaviour, political analysis, and survey design and methodology.

CASSR publishes the findings of research undertaken by Research Associates and staff in three series; The Centre for Applied Social and Survey Research Monograph Series, CASSR Technical Papers and CASSR Research Reports.

Further information is available from The Secretary, Centre for Applied Social and Survey Research, School of Social Sciences, The Flinders University of South Australia, Bedford Park, S.A., 5042.

CASSR TECHNICAL PAPERS

NO. 4

CONSTRUCTING QUESTIONNAIRES WITH RANDOMIZED ORDERING
AND SAMPLING OF ITEMS: A COMPUTER IMPLEMENTED METHOD

Edmund R. Peay
(Senior Lecturer in Psychology,
The Flinders University of South Australia)

The Centre for Applied Social and Survey Research,
School of Social Sciences,
The Flinders University of South Australia.

Copyright 1982 by Edmund R. Peay.

Published 1982 by the Centre for Applied Social & Survey Research
The Flinders University of South Australia, Bedford Park, S.A., 5042.

National Library of Australia
ISBN 0 7258 0201 4

ISSN 0155-9966

CASSR Technical Papers, No. 4,
Edmund R. Peay, 1982

CONSTRUCTING QUESTIONNAIRES WITH RANDOMIZED ORDERING
AND SAMPLING OF ITEMS: A COMPUTER IMPLEMENTED METHOD

\$3.50 (plus postage)

CONSTRUCTING QUESTIONNAIRES WITH RANDOMIZED ORDERING AND SAMPLING OF ITEMS: A COMPUTER IMPLEMENTED METHOD

In constructing a survey or experimental questionnaire, the accomplished researcher devotes a great deal of attention to the selection of items and the order in which they will appear. Item sequencing must take into account a variety of order and contiguity effects. Efforts to deal with these in the course of structuring the questionnaire may not be sufficient, and sometimes attempts may be made to minimize them by more elaborate strategies, such as using alternative forms of the questionnaire incorporating counterbalancing. However, the increase in effort and expense entailed in producing and keeping track of a number of versions of a questionnaire will quickly limit the lengths to which such an approach is carried. It provides only a partial solution in any case. Most often, the researcher will be discouraged by such measures and choose to avoid thinking further about the matter.

In a similar vein, both respondents' impatience and practical considerations will constrain the number of items which can be included in a questionnaire if it is to be effective. Once again, the use of alternative versions of the questionnaire provides a partial answer, but with the same costs and difficulties and with the possibility of item combination as well as order effects.

For the problems of both item sequencing and item inclusion, the adequacy of the solution provided by the use of alternative forms increases with the number of different forms used, but with increasing expense this avenue rapidly becomes prohibitive.

The method for questionnaire construction to be described in this paper makes it convenient to generate as many different forms for a questionnaire as there are respondents. The method is based on using the computer to produce the questionnaire forms themselves¹. In this way the items or subgroups of items of the questionnaire may be randomly ordered or sampled as each form is being generated. This provides a very direct way of dealing with order and combination effects. The technique also provides a number of

incidental benefits, such as direct production of forms bypassing some stages of the usual process, and the prevention of collaborative responses in a group administration situation, since each form is "different".

The following sections will include a description of the method and of a pair of computer programs designed to implement it, a User's Guide for each program, and program listings.

1. In our discussions, the term "questionnaire" will refer to the full document consisting of a total set of items, while "form" will denote a specific generated copy, incorporating a particular reordering and sampling of items. Since such forms will, in general, differ from one another, this terminology is consistent with conventional usage.

The Method

Under this procedure, the individual forms are generated and printed under the control of a computer program which draws upon a stored text file containing the questions and other questionnaire items such as headings and response alternatives. Thus every form may differ in the sequence in which items appear and also in the set of items included.

Of course, this might appear to present a problem for the coding of completed forms for analysis. The task of sorting the responses into a common order while coding them from each form would ordinarily be so arduous and the chances for error so great that the procedure would be entirely impractical. However, there is a straightforward solution. While generating each form, the program can record the identity and sequence of the items appearing on it. The coder then records the responses in the order in which they appear on each form, and a second program is used to process these "scrambled" raw data, sorting the responses into a common order using the previously preserved item sequence information. The result is a set of raw data as they might have been coded from a set of forms with identical structure. The sorting program may also carry out additional processing at the same time, such as structuring the sorted data into a more convenient format.

1. Random item permutation

In order to specify how the items on each form are to be ordered and structured, a convenient notation is required. Let us assume that a set of forms is to be generated for a questionnaire with 5 items, identified by item numbers 1-5. Then

(1,2,3,4,5) or, alternatively, (1-5)

will denote that each form will contain items 1-5 and that the order in which they appear is to be determined for each form so that all possible sequences of the 5 items are equally likely. Then, in addition to generating a set of forms, each identified by a code and containing the 5 items in a different random order, the computer program will preserve, for each form, the order of the items on that form.

2. Mixed ordering with fixed and permuted groups

There will generally be some reason to fix the ordering of certain items or groups of items in a questionnaire; for example, to preserve a logical progression of content, or to avoid contamination from prior information and response sets. One very common contingency calling for fixed ordering even when all or nearly all items of a questionnaire are to be permuted is the necessity for inserting headings and instructions in appropriate places. All but the simplest questionnaires are organized into sections, which call for groupings of items often preceded by section headings. Furthermore, one of the circumstances in which random

permutation may be most worthwhile is the presence of an individual question with a large or complex set of response alternatives, where order effects may be particularly worrisome (see Oppenheim, 1966, p. 56). Such alternatives may be treated as a set of items and randomly permuted, but they must be preceded, of course, by the text of the question itself along with any category headings (see the example in a later section).

The desired sequence of items for most questionnaires will therefore include items and subgroups of items arranged in a composite of fixed and randomly permuted orders. This suggests that the overall structure may be represented as a hierarchy of groupings, where a subgroup of items arranged in either fixed or randomly permuted order occupies a given position within a higher level order. For example, let us consider a questionnaire consisting of 7 questions, divided into 2 sections, with the items in the first section to be presented in a fixed order and those in the second section to be randomly permuted. Thus, Section 1 might contain items: 1.* (section heading and instructions), 1.1, 1.2, 1.3, 1.4; and Section 2 might consist of: 2.*, 2.1, 2.2, 2.3.

The stipulated ordering could be represented by:

[1.*,[1.1,1.2,1.3,1.4],2.*,(2.1,2.2,2.3)]

where square brackets enclose elements that are to be generated in the fixed order specified. This order specification string is equivalent to both:

[1.*,[1.1-1.4],2.*,(2.1-2.3)] and [1.*-1.4,2.*,(2.1-2.3)]

If the relative order of the two sections themselves within the form is to be determined randomly as well, this would be specified by

([1.*,[1.1-1.4]],[2.*,(2.1-2.3)])

This notational scheme provides for any number of levels within the order specification and therefore permits the specification of mixed orders of any level of complexity required. We note also that 2.*-2.3 could represent a single question with 3 response alternatives that we want to appear in a different order on different forms (see the example in a later section).

3. Item sampling

We may also want each form to include only a subset of a given set of items. We can specify that a random sample of size k of the items in a given group is to be chosen for each form by inserting : k in the order specification string following the labels of the items of the group. For example,

[1.*,[1.1-1.4],2.*,(2.1-2.3:2)]

indicates that 2 of the 3 items of Section 2 are to be

randomly selected (and ordered) for each form.

Where each element in a sampled group is a single item, every form will contain the same number of questions (although the selection of questions will, of course, vary). However, under the present scheme an element of a group may be another entire group (or, in fact, a hierarchy of groups). Therefore sampling may be used to determine randomly the appearance or non-appearance on each form of entire groups of questions or sections of a questionnaire. For example²,

[1.*, (1.1-1.4), ([2.*, (2.1-2.3)], [3.*, (3.1-3.4)]:1)]

specifies forms containing 2 sections: Section 1 and either Section 2 or Section 3, selected with equal probability.

2. The item numbering convention used here is a familiar and convenient one, but not essential. The program to be presented later permits item labels to be any sequence of 1-5 characters.

The Programs

This section describes two FORTRAN programs designed to implement the questionnaire generating method described above. They presently operate on a DEC-10 system but are designed to be used on a wide range of systems with minimal adaptation. Following a brief general description of the programs, a User's Guide is provided for each one.

1. PQUEST produces a set of forms using the texts of a set of items incorporating all of the components of a given questionnaire including headings, instructions, questions, and response alternatives. Each generated form contains these items in a sequence specified by an Order Specification String (OSS) using the notation and performing the functions described in earlier sections. This provides for each form to consist of a composite of fixed and randomly permuted groups of items as well as randomly sampled items, groups, and sections. The structure of the text and other necessary inputs are given in the USER'S GUIDE for PQUEST and illustrated in the example following.

It will be noted that the required inputs include a random integer and sequence of numbers to provide a starting point for the generation of random permutations and sampling. As one of its outputs, the program yields a final integer and sequence. If the program is to be executed a number of times to produce the full set of forms for a given questionnaire, it is desirable to use this output from each run to provide the starting point for the next. This ensures the random properties of the series of permutations generated across the entire set of forms. For the first execution, a starting sequence will be constructed if specified, but an integer must still be supplied. Meticulous methodology would suggest the use of a random number table to select this number, although other procedures may prove irresistible.

The entire set of forms is reproducible from a given starting number and sequence. As a protection against the serious consequences of losing the item orders for a set of forms, the starting number and sequence are recorded in the information appearing on the cover page accompanying each set of forms.

2. PQSORT is a program to "unscramble" the responses from each completed form into a common order. PQSORT uses the item order information recorded by PQUEST as it generates each form. The coder enters the responses in the order in which they occur on each form, and no attention need be paid to the fact that the item order and selection will vary. PQSORT will also organize the re-ordered data into a more convenient format if desired.

3. Listings of the programs will be found in this paper, but copies may be obtained by sending a tape (indicate tape format; must be 9-track, 800 bpi) to the author at the Centre.

User's Guide for PQUEST

Input

The information required by PQUEST is provided in 3 files. The contents of each input file will be described under the variable name for the logical unit; the logical unit number (as presently specified) from which it is read follows in parentheses.

LUTXT (2)

This file contains the texts of the items of the questionnaire, including headings, instructions, questions, and response alternatives. Each item consists of a section of text which is to be printed as a single unit (e.g., response alternatives for a question may be included as part of the same item as the question itself if they are always to appear together in the same order; if the response alternatives are to be sampled or permuted, they must be separate items).

Each item may contain any content up to a specified maximum line length, starting in col. 2, on any number of lines. (The line length and other layout parameters are specified in input file LUCOM.) In addition to the text, each item must contain 3 particular elements:

An item label, line 1, cols. 2-6. Any combination of 1-5 characters, except the special characters [] () , : - which are reserved for the OSS. Each item must have a unique item label (blanks are ignored). If any of the characters is *, item identification will always be suppressed in printing (this ordinarily denotes a heading or response alternative).

A carriage control in col. 1 of each line. This controls the line spacing in printing. For all lines except line 1 of each item, it is a standard Fortran carriage control, although only 'blank' (next line) and 0 (skip a line) should be used.

For the 1st line of each item, different conventions are used for the carriage control. A 'blank' inserts a pre-set number (LSPACE) of blank lines before the item (value for LSPACE input in LUCOM; default 3). This is the normal inter-question interval.

A 1 requires that the item always begin a new page (most often used with a section heading).

A 0 causes one line to be skipped. PQUEST ensures that no item is split between pages when printed, and if this carriage control is 0, the item is further assumed to be a non-initial item in a set of successive items which must appear on the same page (usually a question and its response

alternatives).

If the carriage control is >, no lines are skipped before the start of the item.

If the carriage control is 2-9, the pre-item space is set as with 'blank', and space is also allotted following the item (for open-ended responses). The allotted space is $N \times \text{LSPACE}$ ($N=2, \dots, 9$).

Carriage controls should be used to insert line spacing into the printed forms, particularly for long questionnaires, in preference to blank lines in the text, which are included in text length limitations.

Item termination. A termination character (\$) must appear in col. 1 of a line immediately following the item, to indicate its completion. This line should also contain in cols. 2-4 the number of responses to be coded for that item. This is not required by PQUEST, but it is used by PQSORT in sorting responses from completed forms. (For items such as headings, for which there will be no responses, no entry need be made in these columns, and this will be read appropriately by PQSORT as 0.)

To indicate the end of the file, FINISH must appear in cols. 1-6 of the last line.

LUCOM (1)

This file contains the Order Specification String (OSS) which controls the sequence of items on each of the forms to be generated. The OSS follows the conventions and performs the functions discussed in earlier sections. The file also contains values specifying the identification code and numbers of forms to be generated and parameters controlling form layout.

On the 1st line:

Col.	1- 4	Identification code for set of forms (1-4 characters)
	5- 8	Number of 1st form in set to be generated (I4)
	9-12	Number of last form in set to be generated (I4)
	13-16	INIT if new random sequence is to be started with random integer on line following OSS
	18-20	Page length (no. of lines per page) (I3); Default ³ : 59
	22-24	Line length (excluding carriage control) (I3); Default: 72
	28	Page numbering? (2: Yes, 1: No)

3. "Default" values are used if columns are blank or 0.

- (11); Default: Yes
- 32 Item labelling (labels containing * always skipped; 2: Original item labels, 3: Sequential numbering, 1: None) (11); Default: Original
- 35-36 Standard no. of blank lines between items (12); Default: 3

The OSS occupies cols. 1-80 (blanks ignored) of an unspecified number of lines starting with line 2. It consists of groups of item labels (each label corresponding to an item in LUTXT) along with the following control characters:

- , separates item labels within each group.
- identifies a series of adjacent items in LUTXT by its 1st and last items. E.g., 1.1-1.4 may be substituted for 1.1,1.2,1.3,1.4 in the OSS (if these are successive items in LUTXT).
- [] bounds a group of item labels in the OSS and determines that this group of items will appear on each form in the fixed order specified.
- () bounds a group of item labels in the OSS and determines that this group of items will appear on each form in a randomly chosen order.
- :k following the last label in a group (before the closing parenthesis), where k is a positive integer less than the number of labels in the group, denotes that a random sample of k of the items will appear on each form. (If this notation appears in a square-bracketed group, the first k items will appear on every form.)

A group may occupy the position of a single item within another group. Thus, the OSS may consist of one or more nested hierarchies of groups. Among the more common uses for a hierarchically structured OSS are to permute the response alternatives for a given question, to permit a heading or set of instructions to precede a set of permuted or sampled items, and to permute entire sections of a questionnaire or sample them.

The numbers of left and right brackets and parentheses in an OSS must balance. The OSS itself must be enclosed by a pair of brackets or parentheses. A violation of either of these conditions or another error in syntax will produce an error indication and cause program termination.

On the next line (only if INIT appears on the 1st line):

Col. 1-10 A random integer in 'I' format (1-10 digits for DEC-10)

LUVEC (3)

This file contains a random integer and sequence to start the permutations for this run. The file is ordinarily obtained as an output (to the same logical unit) from the previous execution of PQUEST and is used to continue a set of forms for the same questionnaire being generated in a series of executions (see earlier discussion).

For the initial execution, this file is not required for input, but the final integer and sequence will be output to it. This is indicated by the appearance of INIT and a random integer in LUCOM.

On line 1:

Col. 1- 3 Length of sequence (NRAN) (13). Must be an integer greater than or equal to the number of items in LUTXT plus the maximum permitted number of order specification groups (program variable MAXLST; presently 25).
4-13 A random integer in 'I' format (1-10 digits for DEC-10).

On lines 2- :

Col. 1-72 The integers from 1 to NRAN. Format (18I4).

Output

PQUEST produces 4 output files:

- LUQST (7) - contains the set of generated forms, preceded by a cover page containing run information.
- LUGRD (8) - contains the item orders for all generated forms. Used as an input file for PQSORT to re-order responses from completed forms.
- LUVEC (3) - contains final random integer and sequence. Used as input for next run of PQUEST for same questionnaire (see previous section).
- LUERR (6) - used for error messages (and for prompting information if terminal is being used as l.u. 6; see comments at line 35 of PQUEST).

User's Guide for PQSORT

Input

PQSORT requires 3 input files:

LUORD (3)

This file, produced by PQUEST, contains the item orders for the forms to be processed. If these forms have been generated by more than 1 execution of PQUEST, the files from all such runs should be combined. A final line containing FINISH in cols. 1-6 should be added to this file so that a message will be produced if the item order for a form is not found, rather than an end-of-file error.

LUTXT (1)

This is the same as the input text file for PQUEST. Here it provides the number of items and the number of responses coded for each item (see PQUEST User's Guide).

LUDAT (2)

This file contains the raw unsorted data as coded directly from each completed form.

On line 1:

Col. 1-80 Input format, including parentheses. This is the format in which the unsorted responses coded from one form are to be read. Each coded response must be 1 alphanumeric word ('A' format: 1-4 characters). Multiple responses to a single item must be consecutive.

On line 2:

Col. 1-80 Output format, including parentheses. The re-ordered responses from one form will be written in this format. Each response is 1 alphanumeric word, as in the input format, and multiple responses to a single item are consecutive. The output will include responses for every "respondable" item of the questionnaire (whether presented on that form or not). The code for non-presented items is specified by variable NILL in a DATA statement of PQSORT; it is presently 'blank'.

On line 3:

Col. 1- 4 Set identification code for 1st form (as printed by PQUEST)

5- 8 Form number for 1st form

On lines 4- :

Responses from 1st form read according to specified input format.

Repeat for each form on successive lines.

On last line:

Col. 1- 6 FINISH

Output

PQSORT produces 2 output files:

LUSRT (7) - contains the set of re-ordered responses for each form, organized according to the output format specified in LUDAT.

LUMES (6) - receives error messages and information about the number of response sets processed (l.u. 6 may be a terminal).

An Example

Input filesLUTXT:

*TPg University of Eastern Kangaroo Island
 0 Department of Human Intellectual Resources
 0 Social Insight and Personal Achievement Survey

0 On the following pages, you will find a series of questions designed to plumb the depths of your knowledge and the breadth of your opinions. Please answer them as thoughtfully as possible. Do not dwell on any one question; we are interested in your first reactions. Your responses will be treated in the strictest confidence, subject, of course, to the usual qualifications.

\$

11.* Apperceptive Skills Section

\$

1.1 Who is buried in Grant's Tomb? G__

\$

1.2 Why is the sky blue?

0

a. __ They ran out of green

0

b. __ Why not?

0

c. __ In Sydney it isn't

0

d. __ In honour of the astronauts

\$

1.3 I see visions:

0

a. __ Never

0

b. __ Seldom

0

c. __ Sometimes

0

d. __ Often

0

e. __ Always

\$

1.4 In difficult economic periods, people develop various ways of increasing their take-home income. During the past year, how much money have you taken from poor-boxes (to the nearest dollar)?

0

a. __ 1-10

0

b. __ 11-20

0

c. __ 21-50

0

d. __ 51-100

0

e. __ More than 100

\$

1.5 List at least 1 and not more than 5 of your favorite activities which your spouse does not know about. Then rank order these in terms of how upset you think he or she would be if it were discovered. That is, '1' indicates the greatest degree of outrage, '2' the next greatest, and the highest number indicates the least outrage.

0

0

0

0

0

\$ 10

1.6 I feel a great desire to become a duck.

0

a. ☐ Strongly agree

0

b. ☐ Agree

0

c. ☐ Neither agree nor disagree

0

d. ☐ Disagree

0

e. ☐ Strongly disagree

\$ 1

1.7 Please tick all of the colours that you like:

0

a. ☐ Black

0

b. ☐ Auburn

0

c. ☐ Fuchsia

0

d. ☐ Dun

0

e. ☐ Standard Gold

0

f. ☐ Plaid

0

g. ☐ Drab

0

h. ☐ Gravel

0

i. ☐ Royal Gray

0

j. ☐ Burnt Camel

\$ 10

1 2.*

Opinion items

0

For each of the following statements, indicate the extent to which you agree or disagree with it by placing an 'X' in the appropriate column:

0

Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree
-------------------	-------	-------------------------------	----------	----------------------

\$

0

2.1 Women are inferior

-	-	-	-	-
---	---	---	---	---

\$ 1

0

2.2 I never put off until today what I can do tomorrow

-	-	-	-	-
---	---	---	---	---

\$ 1

0

2.3 A penny saved is worth two in the bush

-	-	-	-	-
---	---	---	---	---

\$ 1

0

2.4 The farmer is our friend

-	-	-	-	-
---	---	---	---	---

\$ 1

0

2.5 We are headed for a better future

-	-	-	-	-
---	---	---	---	---

\$ 1

2

2.6 Explain as fully as possible how you feel about carburettors.

\$ 1

*Last Thank you for your cooperation.

\$

FINISH

LUCOM:

XMPL 1 1INIT 59 72 2 2 3
[*TPg,([1.*,1.1,(1.2-1.4),(1.5-1.7)],[2.*, (2.1-2.5:3),2.6]),*Last]
4560690537

Forms generated by
P Q U E S T
Version 1.2, 10-81

E.R. Peay
Centre for Applied Social and Survey Research
The Flinders University of South Australia
Bedford Park, South Australia

Specified item ordering:

[*TPg,([1.*,1.1,(1.2-1.4),(1.5-1.7)],[2.*, (2.1-2.5:5),2.6]),*Last]

1 Forms, numbered from XMPL 1 to XMPL 1

Form layout -

Total no. of text items: 17

Line length: 72

Page length: 59

Blank lines between items: 3

Pages numbered? YES

Item labelling: ORIGINAL

New random sequence started from:

42 4560690537

Simple order

University of Eastern Kangaroo Island
Department of Human Intellectual Resources
Social Insight and Personal Achievement Survey

On the following pages, you will find a series of questions designed to plumb the depths of your knowledge and the breadth of your opinions. Please answer them as thoughtfully as possible. Do not dwell on any one question; we are interested in your first reactions. Your responses will be treated in the strictest confidence, subject, of course, to the usual qualifications.

Apperceptive Skills Section

1.1 Who is buried in Grant's Tomb? G _____

1.4 In difficult economic periods, people develop various ways of increasing their take-home income. During the past year, how much money have you taken from poor-boxes (to the nearest dollar)?

- a. _____ 1-10
- b. _____ 11-20
- c. _____ 21-50
- d. _____ 51-100
- e. _____ More than 100

1.3 I see visions:

- a. _____ Never
- b. _____ Seldom
- c. _____ Sometimes
- d. _____ Often
- e. _____ Always

1.2 Why is the sky blue?

- a. _____ They ran out of green
- b. _____ Why not?
- c. _____ In Sydney it isn't
- d. _____ In honour of the astronauts

1.6 I feel a great desire to become a duck.

- a. _____ Strongly agree
- b. _____ Agree
- c. _____ Neither agree nor disagree
- d. _____ Disagree
- e. _____ Strongly disagree

1.5 List at least 1 and not more than 5 of your favorite activities which your spouse does not know about. Then rank order these in terms of how upset you think he or she would be if it were discovered. That is, '1' indicates the greatest degree of outrage, '2' the next greatest, and the highest number indicates the least outrage.

_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

1.7 Please tick all of the colours that you like:

- a. _____ Black
- b. _____ Auburn
- c. _____ Fuchsia
- d. _____ Dun
- e. _____ Standard Gold
- f. _____ Plaid
- g. _____ Drab
- h. _____ Gravel
- i. _____ Royal Gray
- j. _____ Burnt Camel

Opinion items

For each of the following statements, indicate the extent to which you agree or disagree with it by placing an 'X' in the appropriate column.

	Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree
2.4 The farmer is our friend	—	—	—	—	—
2.2 I never put off until today what I can do tomorrow	—	—	—	—	—
2.3 A penny saved is worth two in the bush	—	—	—	—	—

2.6 Explain as fully as possible how you feel about carburettors.

Thank you for your cooperation.

Other output filesLUORD:

XMPL	1	15														
	1	2	3	6	5	4	8	7	9	10	14	12	13	16	17	

LUVEC:

42																	
	1894762057																
31	12	28	7	8	38	17	34	9	22	20	40	36	11	39	37	2	16
4	21	27	29	41	18	14	13	5	3	25	15	6	10	35	42	26	32
24	23	33	19	30	1												

Program Limits and Technical Information

1. Existing limits and conversions

The present limitations set by PQUEST dimensioning are as follows:

Max. size of text	
Lines:	500
Words/Line:	20
(i.e., 80 char/line)	
Max. no. of items:	150
Max. No. of item groups:	25
(i.e., pairs of paren. or brackets in OSS)	
Max. length. of random seq:	175
Max. length. of OSS:	600
(i.e., no. of complete item labels and control symbols)	

Dimensions may be increased (available memory permitting) by substituting new values for those given, wherever they appear in PQUEST. New values for 501, 151, and 175 are determined in accord with $500+1$, $150+1$, and $150+25$, respectively.

Corresponding adjustments must be made in PQSORT. It will presently process a maximum of 1000 responses per form.

Logical unit numbers for input and output are set in data statements at the beginning of each program and are easily changed.

All symbols used in the OSS and other notational conventions (except for some carriage controls) are also specified in DATA statements and may be modified readily.

2. Random permutations and sampling

Any features of PQUEST that are likely to require adaptation for computer systems other than the DEC-10 are concerned with randomization and have been isolated in the subroutine SHUFFL, which appears at the end. Complete comments are found there. The routine is based on an algorithm described in Knuth (1971, p. 125). Random sampling is accomplished by truncating a randomly permuted vector of elements.

References

Knuth, D.E. Seminumerical algorithms, Vol. 2. Reading, Mass.: Addison-Wesley, 1971.

Oppenheim, A.N. Questionnaire design and attitude measurement. London: Heinemann, 1966.

```

C PQUEST - Program to construct questionnaires and forms with specified
C           and randomly permuted item orderings and item sampling
C           Copyright E. R. Peay, 1981
C           Centre for Applied Social and Survey Research
C           The Flinders University of South Australia
C           Version 1.2 10-81
C Input logical units:  Commands - LUCOM, Text - LUTXT,
C                       Random vector - LUVEC
C Output logical units: Questionnaires - LUQST, Item orders - LUORD,
C                       Random vector - LUVEC, Error messages - LUERR
C Questionnaire format control parameters (set in DATA statement):
C   LGTHL - Length of text line (excluding carriage control)
C   LGTHP - Length of page (no. lines)
C   LSPACE - No. of blank lines between items
C   QDL - Symbol indicating end of item (in cc col of following line)
C   HEAD - Symbol in item id to suppress label
C   IPNO - 2 for page numbers; 1 for none
C   INDQNO - 2 for original item labels, 3 for sequential, 1 for none
C Note: Uses DEC-10 functions and subroutines: FLOAT, IFIX, RAN, SETRAN,
C       SAVRAN
C Equivalents may have to be substituted for other systems.
  DIMENSION TEXT(501,20),LSQST(151),IDIND(150),QST(5),TFILL(150)
  DIMENSION TCC(501),DIGIT(9)
  COMMON /TABLE1/LSTMAT(150,25)/TABLE2/IQORD(150)
  COMMON /TABLE3/QSTID(150,5),NITEMS,LQID
  COMMON /PARS1/LLIST(25),IPLIST(25),LSLIST(25),NLIST,MAXLST
  COMMON /PARS2/NQORD,ISUBL
  COMMON /RANDM/LRAN(170),NRAN,LIT(170),LITJ(170),NIT,IU
  COMMON /LUNO/LUCOM,LUQST,LUERR
  DATA LUCOM,LUTXT,LUVEC,LUQST,LUORD,LUERR /1,2,3,7,8,6/
  DATA LGTHL,LGTHP,LSPACE,QDL,HEAD,ISUBL,B1,B4,MAXLST,LQID
X   /59,72,3,1H$,1H*,100000,1H ,4H ,25,5/
  DATA IPNO,INDQNO/1,0/
  DATA (DIGIT(K),K=1,9)/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
  DATA CCNSKP/1H>/
C INPUT QUESTIONNAIRE IDENTIFICATION, FORM NUMBER RANGE,
C AND FORM LAYOUT PARAMETERS
C PROMPT IF BEING INPUT AT TERMINAL
C   WRITE (LUERR,201)
  READ (LUCOM,101) SETID,MINID,MAXID,INIT,IP,IL,IN,IT,IS
  IF (IP.NE.0) LGTHP=IP
  IF (IL.NE.0) LGTHL=IL
  IF (IN.NE.0) IPNO=IN-1
  IF (IT.NE.0) INDQNO=IT-2
  IF (IS.NE.0) LSPACE=IS
  LGTHLU=IFIX(FLOAT(LGTHL)/4.+0.8)
  IQC=1
  NITEMS=0
12 READ (LUTXT,102) TCC(IQC),(QST(J),J=1,LQID),TFILLT,
X   (TEXT(IQC,J),J=3,LGTHLU)
  IF ((TCC(IQC).EQ.1HF).AND.(QST(1).EQ.1HI).AND.(QST(2).EQ.1HN)
X   .AND.(QST(3).EQ.1HI).AND.(QST(4).EQ.1HS)
X   .AND.(QST(5).EQ.1HH)) GO TO 16
  NITEMS=NITEMS+1
  TFILL(NITEMS)=TFILLT
  DO 13 K=1,LQID
13 QSTID(NITEMS,K)=QST(K)

```

```

      LSQST(NITEMS)=IQC
14  IQC=IQC+1
      READ (LUTXT,104) TCC(IQC),(TEXT(IQC,J),J=1,LGTHLU)
      IF (TCC(IQC).NE.QDL) GO TO 14
      GO TO 12
16  LSQST(NITEMS+1)=IQC
      WRITE (LUQST,210)
      CALL TABLE
      DO 18 K=1,NITEMS
18  IDIND(K)=0
C READ OR INITIALIZE RANDOM VECTOR AND INTEGER
      IF (INIT.EQ.4HINIT) GO TO 19
      READ (LUVEC,120) NRAN,IU
      READ (LUVEC,121) (LRAN(K),K=1,NRAN)
      GO TO 21
19  NRAN=NITEMS+MAXLST
      READ (LUCOM,122) IU
      DO 20 K=1,NRAN
20  LRAN(K)=K
21  DO 22 K=1,NRAN
22  LITJ(K)=0
      NIT=0
      DO 26 J=1,NLIST
      DO 26 I=1,LLIST(J)
      ISYM=LSTMAT(I,J)
      IF (ISYM.GE.ISUBL) GO TO 25
      IDIND(ISYM)=IDIND(ISYM)+1
25  NIT=NIT+1
      LIT(NIT)=ISYM
      LITJ(NIT)=J
26  CONTINUE
      DO 32 IQC=1,NITEMS
      IND=IDIND(IQC)
      IF (IND-1) 30,32,28
28  WRITE (LUERR,204) (QSTID(IQC,J),J=1,LQID)
      STOP
30  WRITE (LUERR,206) (QSTID(IQC,J),J=1,LQID)
32  CONTINUE
C GENERATE QUESTIONNAIRES
      K=MAXID-MINID+1
      WRITE (LUQST,211) K,SETID,MINID,SETID,MAXID
      WRITE (LUQST,212) NITEMS,LGTHL,LGTHP,LSPACE
      WRITE (LUQST,213)
      IF (IPNO) 33,33,34
33  WRITE (LUQST,214)
      GO TO 35
34  WRITE (LUQST,215)
35  WRITE (LUQST,216)
      IF (INDQNO) 36,37,38
36  WRITE (LUQST,217)
      GO TO 39
37  WRITE (LUQST,218)
      GO TO 39
38  WRITE (LUQST,219)
39  IF (INIT.EQ.4HINIT) WRITE (LUQST,220) NRAN,IU
      IF (INIT.NE.4HINIT) WRITE (LUQST,221) NRAN,IU,(LRAN(K),K=1,NRAN)
      DO 76 ID=MINID,MAXID
      WRITE (LUQST,222) SETID,ID

```

```

CALL FORDER
WRITE (LUORD,224) SETID,ID,NQORD
WRITE (LUORD,226) (IQORD(K),K=1,NQORD)
NPNO=1
LNO=1
KSEQ=0
DO 76 KC=1,NQORD
K=IQORD(KC)
TCCLK=TCC(LSQST(K))
LBSP=LSPACE
LASP=0
IF (TCCLK.EQ.1H ) GO TO 47
IF (TCCLK.NE.1H1) GO TO 41
IF (LNO.GT.(1)) LNO=LGTHP+1
GO TO 46
41 IF ((TCCLK.NE.1H0).AND.(TCCLK.NE.CCNSKP)) GO TO 42
LBSP=0
IF (TCCLK.EQ.CCNSKP) GO TO 46
GO TO 47
42 DO 44 ICCLK=2,9
IF (TCCLK.EQ.DIGIT(ICCLK)) GO TO 45
44 CONTINUE
WRITE (LUERR,208) TCCLK
STOP
45 LASP=ICCLK*LSPACE
46 TCCLK=1H
47 LQ=LBSP+LASP
IF (TCCLK.EQ.1H0) GO TO 49
DO 48 KB=KC,NQORD
KT=IQORD(KB)
LB=LSQST(KT)
IF ((KT.NE.K).AND.(TCC(LB).NE.1H0)) GO TO 49
LT=LSQST(KT+1)-1
DO 48 IQC=LB,LT
IF (TCC(IQC).EQ.1H0) LQ=LQ+1
48 LQ=LQ+1
49 LNO=LNO+LQ
IH=0
DO 59 J=1,LQID
IF (QSTID(K,J).EQ.HEAD) IH=1
59 CONTINUE
IF ((LNO.LE.LGTHP).AND.(TCCLK.NE.1H1)) GO TO 64
LNO=1+LQ
NPNO=NPNO+1
IF (IPNO.NE.(1)) GO TO 62
WRITE (LUQST,228) NPNO
GO TO 64
62 WRITE (LUQST,228)
64 IF (LBSP.EQ.(0)) GO TO 67
DO 66 I=1,LBSP
66 WRITE (LUQST,230)
67 IF (IH.EQ.(1)) GO TO 68
IF (INDQNO ) 68,69,70
68 WRITE (LUQST,232) TCCLK,B4,B1,TFILL(K),
X (TEXT(LSQST(K),J),J=3,LGTHLU)
GO TO 72
69 WRITE (LUQST,233) TCCLK,(QSTID(K,J),J=1,LQID),TFILL(K),
X (TEXT(LSQST(K),J),J=3,LGTHLU)

```

```

GO TO 72
70 KSEQ=KSEQ+1
   WRITE (LUQST,234) TCCLK,KSEQ,TFILL(K),
X       (TEXT(LSQST(K),J),J=3,LGTHLU)
72 IF ((LSQST(K+1)-LSQST(K)).LE.(1)) GO TO 74
   LB=LSQST(K)+1
   LT=LSQST(K+1)-1
   DO 73 I=LB,LT
73 WRITE (LUQST,235) TCC(I),(TEXT(I,J),J=1,LGTHLU)
74 IF (LASP.EQ.(0)) GO TO 76
   DO 75 I=1,LASP
75 WRITE (LUQST,230)
76 CONTINUE
   REWIND LUVEC
   WRITE (LUVEC,120) NRAN,IU
   WRITE (LUVEC,121) (LRAN(K),K=1,NRAN)
101 FORMAT (A4,2I4,A4,2(1X,I3),2(3X,I1),2X,I2)
102 FORMAT (A1,5A1,A3,18A4)
104 FORMAT (A1,20A4)
120 FORMAT (I3,I)
121 FORMAT (18I4)
122 FORMAT (I)
201 FORMAT (1H ,44HSet ID, 1st form no., last form no. (A4,2I4)/
X       1H ,47H(also optional layout parameters (4I4)) : >,$)
204 FORMAT (1H ,38HITEM APPEARS MORE THAN ONCE IN ORDER: ,5A1)
206 FORMAT (1H ,45HWARNING ONLY: ITEM DOES NOT APPEAR IN ORDER :,
X       6A1)
208 FORMAT (1H ,30HUNRECOGNIZED SPACING CONTROL: ,A1)
210 FORMAT (1H ,18HForms generated by,9X,9HE.R. Peay/
X       1H ,11HP Q U E S T,16X,
X       45HCentre for Applied Social and Survey Research/
X       1H ,18HVersion 1.2, 10-81,9X,
X       42HThe Flinders University of South Australia/
X       1H ,27X,29HBedford Park, South Australia/////
X       1H ,24HSpecified item ordering:/)
211 FORMAT (1H0,I4,22H Forms, numbered from ,A4,I4,4H to ,A4,I4)
212 FORMAT (1H0,14HForm layout - /
X       28H Total no. of text items: ,I3/16H Line length: ,I3/
X       16H Page length: ,I3/
X       30H Blank lines between items: ,I2)
213 FORMAT (1H ,18HPages numbered? )
214 FORMAT (1H+,18X,2HNO)
215 FORMAT (1H+,18X,3HYES)
216 FORMAT (1H ,18HItem labelling: )
217 FORMAT (1H+,18X,4HNONE)
218 FORMAT (1H+,18X,8HORIGINAL)
219 FORMAT (1H+,18X,20HSEQUENTIAL NUMBERING)
220 FORMAT (/////1H0,33HNew random sequence started from:/
X       1H ,I3,I,3X/1H ,12HSimple order)
221 FORMAT (/////1H0,35HOld random sequence continued from:/
X       1H ,I3,I/(1H ,18I4))
222 FORMAT (1H1,A4,I4)
224 FORMAT (A4,2I4)
226 FORMAT (18I4)
228 FORMAT (1H1,70X,I2)
230 FORMAT (1H )
232 FORMAT (A1,A4,A1,A3,18A4)
233 FORMAT (A1,5A1,A3,18A4)

```

```

234 FORMAT (A1,I5,A3,18A4)
235 FORMAT (A1,20A4)
STOP
END

```

C
C
C

C TABLE - SETS UP HIERARCHICAL LIST STRUCTURE

```

SUBROUTINE TABLE
DIMENSION MPOP(25)
COMMON /TABLE1/LSTMAT(150,25)
COMMON /TABLE3/QSTID(150,5),NITEMS,LQID
COMMON /PARS1/LLIST(25),IPLIST(25),LSLIST(25),NLIST,MAXLST
COMMON /PARS2/NQORD,ISUBL
COMMON /PERM1/CSTR(80),LCSTR,ISTR(600),LISTR,NBRAC
COMMON /SYMBOL/ICOMMA,ICOLON,IDASH,IOP,ICP,IOB,ICB
COMMON /DPARAM/KDSTR
COMMON /LUNO/LUCOM,LUQST,LUERR
DATA LCSTR/80/
DATA ICOMMA,ICOLON,IDASH,IOP,ICP,IOB,ICB/-1,-2,-3,-4,-5,-6,-7/

```

C READ AND CONVERT ORDER SPECIFICATION

```

NBRAC=0
LISTR=0
ISYML=0
ISYMU=0
KDSTR=0
12 READ (LUCOM,102) (CSTR(K),K=1,LCSTR)
WRITE (LUQST,200) (CSTR(K),K=1,LCSTR)
CALL NUMBER
IF (NBRAC) 991,16,12

```

C SET UP HIERARCHICAL LIST STRUCTURE FROM ORDER SPECIFICATION STRING

```

16 DO 22 J=1,MAXLST
   LLIST(J)=0
   IPLIST(J)=0
22 LSLIST(J)=0
   NLIST=0
   KC=0
   LPOP=0
   IF ((ISTR(1).NE.IOP).AND.(ISTR(1).NE.IOB)) GO TO 992
   JC=0
24 KC=KC+1
   IF (KC.GT.LISTR) GO TO 991
   ISYM=ISTR(KC)
   IF (ISYM) 28,992,26

```

C ITEM NUMBER ENCOUNTERED

```

26 LSTMAT(IC,JC)=ISYM
GO TO 24
28 IF (ISYM.NE.IDASH) GO TO 32

```

C DASH ENCOUNTERED

```

   IF ((KC.EQ.(1)).OR.(KC.EQ.LISTR).OR.(ISTR(KC-1).LE.(0))
X    .OR.(ISTR(KC+1).LE.(0))) GO TO 992
   KC=KC+1
   ISYML=ISTR(KC-2)+1
   ISYMU=ISTR(KC)
   IF ((ISYMU-ISYML).LT.(0)) GO TO 992
   DO 30 K=ISYML,ISYMU
   IC=IC+1
30 LSTMAT(IC,JC)=K

```

```

      GC TO 24
32 IF (ISYM.NE.ICOMMA) GO TO 34
C COMMA ENCOUNTERED
      IC=IC+1
      GO TO 24
34 IF (ISYM.NE.ICOLON) GO TO 37
C COLON ENCOUNTERED
      IF ((KC.GE.(LISTR-1)).OR.(KC.LT.(5)).OR.(.NOT.((ISTR(KC+2).EQ.ICP)
X      .OR.(ISTR(KC+2).EQ.ICB)))) GO TO 992
      KC=KC+1
      ISYM=ISTR(KC)
      IF (ISYM) 992,992,36
36 LSLIST(JC)=ISYM
      GO TO 24
37 IF ((ISYM.NE.ICP).AND.(ISYM.NE.ICB)) GO TO 38
C CLOSE PARENTHESIS OR BRACKET ENCOUNTERED
      IF (KC.LT.(3)) GO TO 992
      IF (KC.LT.LISTR) ISYMU=ISTR(KC+1)
      IF ((KC.LT.LISTR).AND.(.NOT.((ISYMU.EQ.ICOMMA).OR.(ISYMU.EQ.ICP)
X      .OR.(ISYMU.EQ.ICB)))) GO TO 992
      IF (ISTR(KC-1).EQ.ICOMMA) GO TO 992
      LLIST(JC)=IC
      LPOP=LPOP-1
      IF (LPOP.LE.(0)) GO TO 52
      JC=MPOP(LPOP)
      IC=LLIST(JC)
      GO TO 24
38 IF ((ISYM.NE.IOP).AND.(ISYM.NE.IOB)) GO TO 992
C OPEN PARENTHESIS OR BRACKET ENCOUNTERED
      IF (KC.GT.LISTR-2) GO TO 992
      IF (KC.GT.1) ISYML=ISTR(KC-1)
      IF ((KC.GT.1).AND.(.NOT.((ISYML.EQ.ICOMMA).OR.(ISYML.EQ.IOP)
X      .OR.(ISYML.EQ.IOB)))) GO TO 992
      IF (ISTR(KC+1).EQ.ICOMMA) GO TO 992
      NLIST=NLIST+1
      IF (NLIST.LE.MAXLIST) GO TO 42
      WRITE (LUERR,206)
      STOP
42 IF (JC.EQ.(0)) GO TO 44
      LSTMAT(IC,JC)=ISUBL+NLIST
      LLIST(JC)=IC
44 LPOP=LPOP+1
      MPOP(LPOP)=NLIST
      JC=NLIST
      IC=1
C OPEN PARENTHESIS ENCOUNTERED
      IF (ISYM.EQ.IOP) IPLIST(JC)=1
      GO TO 24
52 IF ((LPOP.NE.(0)).OR.(KC.NE.LISTR)) GO TO 991
      DO 54 J=1,NLIST
      IF (LSLIST(J).GT.LLIST(J)) GO TO 992
      IF (LSLIST(J).EQ.(0)) LSLIST(J)=LLIST(J)
54 CONTINUE
      RETURN
991 WRITE (LUERR,202)
      STOP
992 WRITE (LUERR,204)
      STOP

```

```

102 FORMAT (80A1)
200 FORMAT (1H ,80A1)
202 FORMAT (1H ,49HORDER COMMAND INCORRECT: BRACK OR PAREN UNMATCHED)
204 FORMAT (1H ,43HORDER COMMAND INCORRECT: IMPROPER SYMBOL OR,
X      10H PLACEMENT)
206 FORMAT (1H ,47HTOO MANY ITEM GROUPS IN OSS - CHANGE DIMENSIONS)
END

```

C
C
C

C NUMBER - CONVERTS ORDER SPEC STRING TO STRING OF INTEGERS
C ITEM LABELS REPRESENTED BY LOCATION IN ITEM ID LIST

```

SUBROUTINE NUMBER
DIMENSION DSTR(5),SYMB(7)
COMMON /PERM1/CSTR(80),LCSTR,ISTR(600),LISTR,NBRAC
COMMON /TABLE3/QSTID(150,5),NITEMS,LQID
COMMON /SYMBOL/ICOMMA,ICOLON,IDASH,IOP,ICP,IOB,ICB
COMMON /DPARAM/KDSTR
COMMON /LUNO/LUCOM,LUQST,LUERR
DATA NSYMB,(SYMB(K),K=1,7),BLANK/7,
X      1H,,1H:,1H-,1H(,1H),1H[,1H],1H /
      KCSTR=0
22 KCSTR=KCSTR+1
   IF (KCSTR.GT.LCSTR) RETURN
   SYMT=CSTR(KCSTR)
   IF (SYMT.EQ.BLANK) GO TO 22
   DO 24 KC=1,NSYMB
   IF (SYMT.EQ.SYMB(KC)) GO TO 32
24 CONTINUE
   KDSTR=KDSTR+1
   IF (KDSTR.GT.LQID) GO TO 49
   DSTR(KDSTR)=SYMT
   GO TO 22
32 KC=-KC
   IF ((KC.EQ.IOP).OR.(KC.EQ.IOB)) NBRAC=NBRAC+1
   IF ((KC.EQ.ICP).OR.(KC.EQ.ICB)) NBRAC=NBRAC-1
   IF (KDSTR.EQ.(0)) GO TO 51
   IF (LISTR.EQ.(0)) GO TO 42
   IF (ISTR(LISTR).NE.ICOLON) GO TO 42
   IQC=INTNO(DSTR,KDSTR)
   GO TO 50
42 DO 48 IQC=1,NITEMS
   KID=0
   DO 45 KP=1,KDSTR
   SYMT=DSTR(KP)
44 KID=KID+1
   IF (KID.GT.LQID) GO TO 48
   SYMID=QSTID(IQC,KID)
   IF (SYMID.EQ.BLANK) GO TO 44
   IF (SYMID.NE.SYMT) GO TO 48
45 CONTINUE
   IF (KID.EQ.LQID) GO TO 50
   KT=KID+1
   DO 46 KID=KT,LQID
   IF (QSTID(IQC,KID).NE.BLANK) GO TO 48
46 CONTINUE
   GO TO 50
48 CONTINUE

```

```

49 IF (KDSTR.GT.LQID) KDSTR=LQID
   WRITE (LUERR,202) (DSTR(KP),KP=1,KDSTR)
   STOP
50 LISTR=LISTR+1
   ISTR(LISTR)=IQC
   KDSTR=0
51 LISTR=LISTR+1
   ISTR(LISTR)=KC
   GO TO 22
202 FORMAT (1H ,19HUNKNOWN ITEM LABEL,
X          38H OR TOO LARGE INTEGER, IN ORDER SPEC: ,5A1/
X          1H ,43H(MAY INDICATE ORDER SPEC STRING NOT CLOSED))
   END

```

C
C
C

```

C FORDER - PERMUTES APPROPRIATE LISTS AND SETS UP ITEM ORDER
  SUBROUTINE FORDER
    DIMENSION MPOP(25),MLIST(25)
    COMMON /TABLE1/LSTMAT(150,25)/TABLE2/IQORD(150)
    COMMON /PARS1/LLIST(25),IPLIST(25),LSLIST(25),NLIST,MAXLST
    COMMON /PARS2/NQORD,ISUBL
    COMMON /RANDM/LRAN(170),NRAN,LIT(170),LITJ(170),NIT,IU
    CALL SHUFFL(LRAN,NRAN,IU)
    DO 12 J=1,NLIST
12  MLIST(J)=0
    IC=0
    DO 22 KC=1,NRAN
      JC=LITJ(LRAN(KC))
      IF (JC.EQ.(0)) GO TO 22
      IC=IC+1
      IF (IPLIST(JC).EQ.(0)) GO TO 22
      ISYM=LIT(LRAN(KC))
      MLIST(JC)=MLIST(JC)+1
      LSTMAT(MLIST(JC),JC)=ISYM
      IF (IC.EQ.NIT) GO TO 24
22  CONTINUE
24  CONTINUE
    NQORD=0
    LPOP=1
    MPOP(1)=1
    JC=1
    IC=0
32  IC=IC+1
    IF (IC.GT.LSLIST(JC)) GO TO 34
    KC=LSTMAT(IC,JC)
    IF (KC.GE.ISUBL) GO TO 36
    NQORD=NQORD+1
    IQORD(NQORD)=KC
    GO TO 32
34  LPOP=LPOP+1
    IF (LPOP.EQ.(0)) GO TO 42
    JC=MPOP(LPOP)
    IC=MLIST(JC)
    GO TO 32
36  MPOP(LPOP)=JC
    MLIST(JC)=IC
    LPOP=LPOP+1

```

```

C PQSORT - Program to sort responses from forms generated by PQUEST
C Input logical units: No. resp. for each item (from text file) - LUTXT
C                      (A1,A3)
C                      Item orders - LUORD; in- and out-formats,
C                      (18I4)                      (80A1)
C                      responses - LUDAT
C Output logical units: Format, reordered responses - LUSRT;
C                      Messages - LUMES
C Note: Each response is 1 alphanumeric word (1-4 characters),
C       in both input and output variable formats
C       DIMENSION NRT(150),NRTCUM(151),DFORM(80),SFORM(80)
C       DIMENSION IQORD(150),IQRESP(1000),IQSRT(1000),DRT(3)
C       DATA LUTXT,LUORD,LUDAT,LUSRT,LUMES/1,3,2,7,6/
C       DATA SNILL,QDL,LRT/4H      ,1H$,3/
C READ (FROM TEXT FILE) AND CUMULATE NOS. OF RESPONSES FOR TOTAL SET
C OF ITEMS (STORED ONE POSITION UP)
  KC=0
10 READ (LUTXT,104) QLDT,(DRT(K),K=1,LRT)
  IF (QLDT.EQ.1HF) GO TO 12
  IF (QLDT.NE.QDL) GO TO 10
  KC=KC+1
  NRT(KC)=INTNO(DRT,LRT)
  GO TO 10
12 NQST=KC
  NRTCUM(1)=0
  LT=NQST+1
  DO 14 KC=2,LT
14  NRTCUM(KC)=NRTCUM(KC-1)+NRT(KC-1)
  NRTOT=NRTCUM(NQST+1)
  READ (LUDAT,106) (DFORM(J),J=1,80)
  READ (LUDAT,106) (SFORM(J),J=1,80)
  NSETS=0
C READ AND REORDER EACH SET OF RESPONSES
22 READ (LUDAT,108) SETIDD,IDD
  IF (SETIDD.EQ.4HFINI) GO TO 99
  NSETS=NSETS+1
C  IF A TERMINAL IS BEING USED, GIVE RUNNING INFORMATION
C  WRITE (LUMES,202) NSETS,SETIDD,IDD
C  FIND RESPONSE ORDER FOR THIS SET
  REWIND LUORD
24 READ (LUORD,112) SETIDO,IDO,NQORD
  IF (SETIDO.NE.4HFINI) GO TO 26
  WRITE (LUMES,114) SETIDD,IDD
  STOP
26 IF ((SETIDD.NE.SETIDO).OR.(IDD.NE.IDO)) GO TO 24
  READ (LUORD,116) (IQORD(KC),KC=1,NQORD)
C FIND TOTAL NUMBER OF RESPONSES FOR THIS SET
  NRESP=0
  DO 28 KC=1,NQORD
28  NRESP=NRESP+NRT(IQORD(KC))
  READ (LUDAT,DFORM) (IQRESP(KDAT),KDAT=1,NRESP)
C  RELOCATE RESPONSES FOR EACH ITEM
  DO 31 K=1,NRTOT
31  IQSRT(K)=SNILL
  KDAT=0
  DO 32 KC=1,NQORD
  K=IQORD(KC)
  IF (NRT(K).EQ.(0)) GO TO 32

```

```

JC=KC-ISUBL
IC=0
GO TO 32
42 CONTINUE
RETURN
END

```

C
C
C

```

C INTNO - CONVERTS AN ALPHA STRING TO AN INTEGER
FUNCTION INTNO(DSTR,KDSTR)
  DIMENSION DSTR(5),DIGIT(11)
  DATA (DIGIT(KD),KD=1,11)/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0,
X      1H /
  INTNO=0
  DO 16 K=1,KDSTR
    DT=DSTR(K)
    DO 12 KD=1,11
      IF (DIGIT(KD).EQ.DT) GO TO 14
12 CONTINUE
      WRITE (6,202) (DSTR(KD),KD=1,KDSTR)
      STOP
14 IF (KD.GE.(10)) KD=0
16 INTNO=INTNO+KD*10**(KDSTR-K)
  RETURN
202 FORMAT (1H ,40HSTRING ASSUMED TO BE AN INTEGER IS NOT: ,10A5)
END

```

C
C
C

```

C SHUFFL - SUBROUTINE TO PRODUCE RANDOM PERMUTATIONS
C*****USES DEC-10 RANDOM NUMBER GENERATOR 'RAN' WHICH PRODUCES
C*****0<=X<1.0, AND SUBROUTINES 'SETRAN' AND 'SAVRAN'. 'SETRAN(IU)'
C*****PROVIDES A STARTING VALUE IU (INTEGER <2**32) FOR 'RAN'.
C*****'SAVRAN(IU)' SAVES THE LAST NUMBER GENERATED BY 'RAN' IN IU.
C*****'SHUFFL' MAY HAVE TO BE ADAPTED FOR SOME OTHER COMPUTER SYSTEMS.
C SUBROUTINE BASED ON AN ALGORITHM DESCRIBED IN KNUTH, D.E.
C SEMINUMERICAL ALGORITHMS, VOL. 2. ADDISON-WESLEY, 1971. P. 125
C IVEC: VECTOR TO BE "SHUFFLED"; N: SIZE OF VECTOR; IU: RANDOM NUMBER
C SAVE IU AND IVEC BETWEEN RUNS; SEQUENCE IS REPRODUCIBLE
C FROM IVEC AND IU
  SUBROUTINE SHUFFL(IVEC,N,IU)
    INTEGER IVEC(N),T
    CALL SETRAN(IU)
    J=N
    DO 10 I=1,N-1
      K IS A RANDOM INTEGER 1<=K<=J
      K=IFIX(RAN(0)*FLOAT(J))+1
      T=IVEC(J)
      IVEC(J)=IVEC(K)
      IVEC(K)=T
      J=J-1
10 CONTINUE
    CALL SAVRAN(IU)
    RETURN
  END

```

```

      LRK=NRTCUM(K+1)-NRTCUM(K)
      KSRT=NRTCUM(K)
      DO 32 XT=1,LRK
      KSRT=KSRT+1
      KDAT=KDAT+1
      IQSRT(KSRT)=IQRESP(KDAT)
32  CONTINUE
      WRITE (LUSRT,108) SETIDD,IDD
      WRITE (LUSRT,SFORM) (IQSRT(KC),KC=1,NRTOT)
      GO TO 22
99  WRITE (LUMES,204) NSETS
      STOP
104  FORMAT (A1,3A1)
106  FORMAT (80A1)
108  FORMAT (A4,A4)
112  FORMAT (A4,A4,I4)
114  FORMAT (1H ,39HItem order not found for response set: ,A4,I4)
116  FORMAT (18I4)
202  FORMAT (1H ,I4,1H.,2X,A4,A4)
204  FORMAT (/1H ,I4,24H Response sets processed)
      END

```

C
C
C

```

C  INTNO - CONVERTS AN ALPHA STRING TO AN INTEGER
      FUNCTION INTNO(DSTR,KDSTR)
      DIMENSION DSTR(10),DIGIT(11)
      DATA (DIGIT(KD),KD=1,11)/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0,
X                                     1H /
      INTNO=0
      DO 16 K=1,KDSTR
      DT=DSTR(K)
      DO 12 KD=1,11
      IF (DIGIT(KD).EQ.DT) GO TO 14
12  CONTINUE
      WRITE (6,202) (DSTR(KD),KD=1,KDSTR)
      STOP
14  IF (KD.GE.(10)) KD=0
16  INTNO=INTNO+KD*10**(KDSTR-K)
      RETURN
202  FORMAT (1H ,40HSTRING ASSUMED TO BE AN INTEGER IS NOT: ,10A5)
      END

```

The Centre for Applied Social and Survey Research (CASSR) was established in the School of Social Sciences in The Flinders University of South Australia in 1977. Its aims are to encourage and co-ordinate multi-disciplinary research in the social sciences, particularly in the areas of social survey research, urban and regional planning and development, community studies, human behaviour, social policy, and political analysis. Members of the academic staff of the University working in these and related areas may become Research Associates in the CASSR. Research conducted in the CASSR is funded both from the University Research Budget and from a wide variety of external sources. Non-academic staff employed in the CASSR are funded from external grants. The affairs of the CASSR are administered by a Management Committee appointed by the University Council to which it reports annually.

The CASSR publishes from time to time research undertaken in the Centre. Three series are produced: a Monograph Series, which are substantial pieces of research published on the recommendation of an Editorial Board; CASSR Research Papers, reporting on the results of on-going research activities; and Technical Papers, relating to survey research methodology, data systems design and data analysis.

The responsibility for the opinions expressed in the CASSR publications rests with the authors and does not necessarily reflect the opinions of the CASSR Management Board or the University.

Enquiries concerning circulation of and orders for these publications should be addressed to: The Secretary, Centre for Applied Social and Survey Research (CASSR), School of Social Sciences, The Flinders University of South Australia, Bedford Park, Sth. Aust., Australia, 5042. Telephone (08) 275 2396.

CASSR PUBLICATIONS

Centre for Applied Social & Survey Research Monograph Series:

- NO. 1 URBAN SOUTH AUSTRALIA: SELECTED READINGS
— edited by Clive A. Forster & Robert J. Stimson
(November, 1977) \$6.50*
- NO. 2 SUBURBAN HEALTH CARE BEHAVIOUR IN ADELAIDE
— E.A. Cleland, R.J. Stimson, A.J. Goldsworthy
(December, 1977) \$9.50*
- NO. 3 UNIVERSITIES & GRADUATE EMPLOYMENT: ANTICIPATING
THE '80s: PROCEEDINGS OF A SYMPOSIUM
— edited by E.A. Cleland
(November, 1979) \$6.00*
- NO. 4 THE FUTURE ROLE OF LOCAL GOVERNMENT IN SOUTH AUSTRALIA:
MANAGEMENT SYSTEMS AND DATA REQUIREMENTS: PROCEEDINGS
OF A SUMMER SCHOOL
— edited by T.K. Bell, C.A. Forster, G.B. Mitchell & R.J. Stimson
(December, 1979) \$7.00*

CASSR Technical Papers:

- NO. 1 ACCEPTANCE OF THE NULL HYPOTHESIS IN EXPERIMENTAL AND
SURVEY RESEARCH
— Edmund R. Peay
(February, 1978) \$1.00*
- NO. 2 SYMBOL DESIGN & TESTING METHODOLOGY PROJECT: DETERMINATION
OF NEED FOR A SYMBOL
— David Sless & Peter Cairney
(September, 1978) \$1.00*
- NO. 3 SYMBOL DESIGN & TESTING METHODOLOGY PROJECT: A PROPOSED
TESTING METHODOLOGY FOR PUBLIC INFORMATION SYMBOLS \$1.00*
- NO. 4 CONSTRUCTING QUESTIONNAIRES WITH RANDOMIZED ORDERING
AND SAMPLING OF ITEMS: A COMPUTER IMPLEMENTED METHOD
— Edmund R. Peay
(April, 1982) \$3.50*

*Plus postage

Publications are available from:

The Secretary,
Centre for Applied Social & Survey Research (CASSR),
School of Social Sciences,
The Flinders University of South Australia,
BEDFORD PARK, S.A., 5042.